IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT:	Anthony F. Gigliotti	CONFIRMATION NO.:	5015
APPLICATION NO.:	10/763,814		
FILING DATE:	January 22, 2004		
TITLE:	DISTRIBUTED POLICY DRIVEN SOFTWARE DELIVERY		
EXAMINER:	Vo, Ted T	TELEPHONE:	(571) 272-3706
		FAX:	(571) 273-8300
ART UNIT:	2191		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CORRECTED APPEAL BRIEF

Dear Sir:

This paper is in support of a Notice of Appeal filed March 31, 2009, of the Office Action dated October 1, 2008, to the Board of Patent Appeals and Interferences. This paper is further responsive to a Notice of Non-compliant Appeal Brief mailed November 20, 2009.

Table of Contents

Real Party in Interest

Autonomic Software, Inc.

Related Appeals and Interferences

None.

Status of Claims

Claims 1-15 and 20-36 have been finally rejected and are on appeal.

Claims 16-19 have been cancelled without prejudice or disclaimer of the subject matter contained therein.

Status of Amendments

No amendments after final have been filed. All amendments have been entered.

Summary of Claimed Subject Matter

The claimed subject matter relates to software updates. Specifically, the claimed subject matter relates to distributed policy driven software delivery. (¶[0001]). A system scans various reporting services and application manufacturers' websites for recent security upgrades, hot fixes, and service packs. (¶[0007], ll. 1-2). The system then retrieves these patches and automatically applies these patches on every computer within the corporate network. (¶[0007], ll. 2-4). By inoculating systems before viruses are able to take advantage of their weaknesses, corporations can prevent many of the modern viruses from entering their network and reduce their corporate losses. (¶[0007], ll. 4-6). Furthermore, as a sufficient amount of network and system administrator time is currently utilized on keeping track of security fixes, downloading these patches, and applying them across the corporate network, the implementation of this solution saves money and resources. (¶[0007], ll. 6-9).

Claim 1 is directed to a method for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization. The method includes receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication. (FIG. 6, reference numeral 604; ¶ [0026] ll. 10-14). The method also includes comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers. (FIG. 6, reference numeral 608; ¶ [0026] ll. 14-18). The method also includes queuing the update if an update exists for an application controlled by an inoculation client. (FIG. 6, reference numeral

608; ¶ [0026] ll. 18-21). The method also includes receiving a communication from the corresponding inoculation client checking for available distribution jobs. (FIG. 6, reference numeral 610; ¶ [0026] ll. 21-22). The method also includes automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client. (FIG. 6, reference numeral 612; ¶ [0026] ll. 22-24).

Claim 20 is directed to an apparatus for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization. The apparatus includes a means (FIG. 7, reference numeral 700) for receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication. (FIG. 6, reference numeral 604; ¶ [0026] ll. 10-14). The apparatus also includes a means (FIG. 7, reference numeral 702) for comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers. (FIG. 6, reference numeral 608; ¶ [0026] ll. 14-18). The apparatus also includes a means (FIG. 7, reference numeral 704) for queuing the update if an update exists for an application controlled by an inoculation client. (FIG. 6, reference numeral 608; ¶ [0026] ll. 18-21). The apparatus also includes a means (FIG. 7, reference numeral 706) for receiving a communication from the corresponding inoculation client checking for available distribution jobs. (FIG. 6, reference numeral 610; ¶ [0026] ll. 21-22). The apparatus also includes a means (FIG. 7, reference numeral 708) for automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an

application controlled by the corresponding inoculation client. (FIG. 6, reference numeral 612; ¶ [0026] ll. 22-24).

Claim 35 is directed to a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization. The method includes receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication. (FIG. 6, reference numeral 604; ¶ [0026] ll. 10-14). The method also includes comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers. (FIG. 6, reference numeral 608; ¶ [0026] ll. 14-18). The method also includes queuing the update if an update exists for an application controlled by an inoculation client. (FIG. 6, reference numeral 608; ¶ [0026] ll. 18-21). The method also includes receiving a communication from the corresponding inoculation client checking for available distribution jobs. (FIG. 6, reference numeral 610; ¶ [0026] ll. 21-22). The method also includes automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client. (FIG. 6, reference numeral 612; ¶ [0026] ll. 22-24).

Claim 36 is directed to an apparatus for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization. The apparatus

includes a global update repository including software updates from multiple application manufacturers. (¶¶[0016], [0017], and [0027])). The apparatus also includes an inoculation server configured to receive application and system information from one or more inoculation clients installed on a network of devices of a peer-to-peer network controlled by an organization, the receiving performed via peer-to-peer communication. (FIG. 7, reference numeral 700; ¶[0027] ll. 10-15). The inoculation server is also configured to compare the application and system information with application and version information in the global update repository to determine if an update exists for a corresponding application controlled by an inoculation client. (FIG. 7, reference numeral 702; ¶[0027] ll. 15-21). The inoculation server is also configured to queue the update if an update exists for an application controlled by an inoculation client. (FIG. 7, reference numeral 704; ¶[0027] ll. 21-24). The inoculation server is also configured to receive a communication from the corresponding inoculation client checking for available distribution jobs. (FIG. 7, reference numeral 706; ¶[0027] ll. 24-26). The inoculation server is also configured to automatically transmit the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client. (FIG. 7, reference numeral 708; ¶[0027] ll. 26-29).

Grounds of Rejection to be Reviewed on Appeal

Whether Claims 1-15 and 20-36 are unpatentable under 35 U.S.C. § 103(a) over "Understanding Patch and Update Management: Microsoft's Software Update Strategy," Microsoft Corporation, October 2003, pp. i-iii, 1-14 (hereinafter, "Microsoft White Paper") in view of "Microsoft dictionary (or a peer-to-peer network architecture definition)," (hereinafter, "Microsoft Dictionary").

Argument

Rejection of Claims 1-15 and 20-36 under 35 U.S.C. § 103

Independent Claims 1, 20, 35, and 36

Independent claim 1 recites, *inter alia*, receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication. The Examiner contends these features are disclosed by Microsoft White Paper. However, these features are not disclosed in Microsoft White Paper.

Contrary to the Examiner's statement, Microsoft White Paper does not disclose receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication as required by Claim 1. In support of the Examiner's statement, the Examiner refers to the entirety of Microsoft White Paper, which speaks generally about patch and update management in small-, medium-, and large-sized organizations, but says nothing about receiving the application and system information via peer-to-peer communication.

In response, the Examiner states:

> ... It is unclear what point in this argument, whether the argument is that Microsoft does not teach receiving application information from the clients or Microsoft does not teach peer-to-peer. Applicants' argument does not directly to the main functionality of the claims, but is off from the heart of the specification. It should be note that the heart of the specification is to disclose patch management as Microsoft does. It should be noted that Internet provides two-way communication, where a clients within Microsoft can transparently use the Microsoft websites, as shown in p. 6 and p. 7, and p. 14, for providing the information. In p. 7, 'Windows Update", it is for a client who communicates with its server for receiving patch update. With a baseline security Analyzer, it allows users to scan (queue) one or more Windows based computers for common security misconfigurations (p. 7-8). This is two-way communications. On the other hand, a peer-to-peer system according to the Microsoft dictionary is only a

> network of two or more computers that use the same program or type of program to communicate and share the data. Each computer acts like a server to other in the network. Thus, the inclusion of the peer-to-peer in a claim does not make it distinct from a pair of server/client computers, or a cluster of computers communicated in an organization like of Microsoft. Therefore, Microsoft discloses the claimed limitation above.[1]

The Applicant respectfully submits the Examiner's reference to "the heart of the specification" evidences an impermissible attempt to ignore limitations found in the claim. The Examiner has used one reference (Microsoft White Paper) of two references in a rejection under 35 U.S.C. § 103 to allegedly read on the claim limitation "receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication." The Examiner's statement "[i]t is unclear … whether the argument is that Microsoft does not teach receiving application information from the clients or Microsoft does not teach peer-to-peer" impermissibly paraphrases limitations found in Claim 1 into two seemingly unrelated limitations. The Applicant respectfully submits the cited references do not teach or suggest what is *claimed* in Claim 1, e.g. "receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication."

Furthermore, the Examiner's statement "a peer-to-peer system according to the Microsoft dictionary is only a network of two or more computers that use the same program or type of program to communicate and share the data. Each computer acts like a server to other in the network. Thus, the inclusion of the peer-to-peer in a claim does not make it distinct from a pair of server/client computers, or a cluster of computers communicated in an organization like of Microsoft" is contrary to what is understood by those skilled in the art. The Examiner's statement refers to "the Microsoft dictionary." It is unclear exactly what publication the

[1] Office Action mailed February 7, 2008 at ¶ 2.

Examiner intends to refer, since the Examiner has not provided a complete citation for this reference. The Applicant assumes the Examiner intends to refer to the publication *Microsoft Computer Dictionary*, Fifth Edition, Microsoft, © 2002. The *Microsoft Computer Dictionary* reference defines term "peer-to-peer architecture" as follows:

> **peer-to-peer architecture** *n*. A network of two or more computers that use the same program or type of program to communicate and share data. Each computer, or *peer*, is considered equal in terms of responsibilities and each acts as a server to the others in the network. Unlike a client/server architecture, a dedicated file server is not required. However, network performance is generally not as good as under client/server, especially under heavy loads. *Also called*: peer-to-peer network. *See also* peer, peer-to-peer communications, server. *Compare* client/server architecture.[2]

The Examiner has agreed that Microsoft White Paper discloses a client/server architecture. And as can be seen by the above definition of peer-to-peer architecture cited by the Examiner, a client/server architecture (like disclosed in Microsoft White Paper) is clearly differentiated from a peer-to-peer architecture. The inclusion of "peer-to-peer" in the claim *does* make it distinct.

Additionally, the Examiner's statement "It should be noted that a distribution of a piece of software or of patches in a network is not new in the art. It is done commonly in software companies whose clients are frequently attacked by hackers."[3] indicates the Examiner's conclusion of obviousness is based on improper hindsight reasoning. In more detail, the Examiner's use of the verb "is," which is the present tense form of the verb "to be" indicates the Examiner's conclusion of obviousness is based upon knowledge as of the date of the Office Action, which is after the time the claimed invention was made. According to established caselaw,

[2] "Microsoft Computer Dictionary, Fifth Edition, " Microsoft, © 2002, p. 397. (emphasis in original)
[3] Office Action mailed October 1, 2008 at p. 2. (emphasis added)

> [a]ny judgement on obviousness is in a sense necessarily a reconstruction based on hindsight reasoning, but so long as it takes into account only knowledge which was within the level of ordinary skill in the art at the time the claimed invention was made and does not include knowledge gleaned only from applicant's disclosure, such a reconstruction is proper."[4]

As the Examiner's statement failed to take into account only knowledge which was within the level of ordinary skill in the art at the time the claimed invention was made, the Examiner's conclusion of obviousness is based improperly on improper hindsight reasoning.

For the above reasons, the 35 U.S.C. § 103 Rejection of Claim 1 based on Microsoft White Paper in view of Microsoft Dictionary is unsupported by the cited art of record. Thus, a *prima facie* case has not been established and the rejection must be withdrawn.

Claim 1 also recites in part "comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers." This is also not disclosed by Microsoft White Paper in view of Microsoft Dictionary. In support of the Examiner's statement, the Examiner refers to portions of Microsoft White Paper that speak generally about updates from a single application manufacturer (Microsoft). But nowhere does Microsoft White Paper disclose a global update repository that includes updates from *multiple* application manufacturers as required by Claim 1.

The Examiner admits that Microsoft White Paper does not teach a global update repository that includes updates from multiple application manufacturers, but does not provide a

[4] *In re McLaughlin*, 443 F.2d 1392, 1395, 170 USPQ 209, 212 (CCPA 1971) (emphasis added); see also M.P.E.P. § 2145, part X.A.

specific reference where such a limitation is found, instead arguing that Microsoft White Paper mentioned a "future consolidation of a centralized update database."[5] The Examiner further states "it is obvious to the ordinary in the art, that GLOBAL UPDATE REPOSITORY is only a centralized storage for easing management. It is only making integral thus renders obviousness."[6] However, the portion of Microsoft White Paper cited by the Examiner apparently refers to a global update repository for a single vendor's products; it does not disclose a global update repository that includes updates from *multiple* application manufacturers as required by Claim 1.

> The Examiner also states:
>
> Applicants argued Microsoft does not do patching via Peer-To-Peer ... It appears Applicants argued Microsoft does not have a Global Update Repository for patching.[7]

First, the Applicants did not allege Microsoft does not teach a global update repository; the Applicants submit Microsoft does not teach the global update repository including updates from multiple application manufacturers, as required by Claim 1. Second, the Applicants respectfully submit that the Examiner's reference to what Microsoft *could* do, and what particular business corporations *can* provide amounts to mere speculation not supported by the cited art of record. The Examiner also points to page 13 of Microsoft White Paper. However, page 13 of Microsoft White Paper includes a discussion of various Microsoft products but says nothing about the claimed limitation of a global update repository including updates from multiple application manufacturers as required by Claim 1.

[5] Office Action at p. 10.
[6] Office Action at p. 10.
[7] Office Action at p.2.

For the above reasons, the 35 U.S.C. § 103 Rejection of Claim 1 based on Microsoft White Paper in view of Microsoft Dictionary is unsupported by the cited art of record. Thus, a *prima facie* case has not been established and the rejection must be withdrawn.

Claims 2-15

Claims 2-15 depend from Claim 1. Claim 1 being allowable, Claims 2-15 must also be allowable.

Claim 9

Claim 9 recites:

> The method of claim 8, therein the global update repository mines, retrieves, and archives external update information.

The Examiner states:

> ... Microsoft discloses, The method of claim 8, therein said global update repository mines, retrieves, and archives external update information (i.e. Microsoft /Microsoft downloads Web site; or see "consolidate the patches and updates into one repository" (p. 13)).[8]

The Applicants respectfully disagree. Contrary to the Examiner's statement, Microsoft does not disclose that the global update repository mines, retrieves, and archives external update information as required by Claim 9. In support of the Examiner's statement, the Examiner refers to page 13 of Microsoft, which speaks generally about consolidating Microsoft patches and updates into one repository, but says nothing about the required claim limitation. The cited portion of Microsoft does not indicate that it is the global update repository that performs the "consolidating" to which the Examiner refers. Furthermore, the Applicants respectfully submit

[8] Office Action at p. 12.

that the Examiner's attempt to equate the "consolidate" of Microsoft with "mines, retrieves, *and* archives" required by Claim 9, is improper. For this additional reason, the 35 U.S.C. § 103 Rejection of Claim 9 is unsupported by the cited art of record and must be withdrawn.

Additionally, the Applicants made the above argument in the Response mailed November 20, 2007. Considering that the Examiner has not provided any comments or rebuttal to the Applicant's argument, but only restated prior rejections, it can be assumed that the Examiner agrees to the Applicant's arguments and that the Claims are allowable.[9]

Claim 10

Claim 10 recites:

> The method of claim 9, wherein the external update information is mined and retrieved from external security websites.

The Examiner states:

> … Microsoft discloses, The method of claim 9, wherein said external update information is mined and retrieved from external security websites (.e. Microsoft/Microsoft downloads Web site; or see "consolidate the patches and updates into one repository" (p. 13)).[10]

The Applicants respectfully disagree. As Claim 10 depends from Claim 9, the arguments made above with respect to Claim 9 apply here as well.

Contrary to the Examiner's statement, Microsoft does not disclose wherein the external update information is mined and retrieved from external security websites as required by Claim 10. In support of the Examiner's statement, the Examiner refers to page 13 of Microsoft, which says nothing about external security websites, let alone mining and retrieving external update

[9] *In re Herrmann*, *supra*.
[10] Office Action at p. 12.

information from the external security websites. For this additional reason, the 35 U.S.C. § 103 Rejection of Claim 10 is unsupported by the cited art of record and must be withdrawn.

Additionally, the Applicants made the above argument in the Response mailed November 20, 2007. Considering that the Examiner has not provided any comments or rebuttal to the Applicant's argument, but only restated prior rejections, it can be assumed that the Examiner agrees to the Applicant's arguments and that the Claims are allowable.[11]

Claim 11

Claim 11 recites:

> The method of claim 10, wherein the global update repository uses web spiders.

The Examiner states:

> ... Microsoft discloses, The method of claim 10, wherein said global update repository uses web spiders (i.e. Microsoft/Microsoft downloads Web site; or see "consolidate the patches and updates into one repository" (p. 13)).[12]

The Applicants respectfully disagree. As Claim 11 depends from Claim 10, the arguments made above with respect to Claim 10 apply here as well.

Contrary to the Examiner's statement, Microsoft does not disclose wherein the global update repository uses web spiders as required by Claim 11. In support of the Examiner's statement, the Examiner refers to page 13 of Microsoft, which speaks generally about patch consolidation but says nothing about web spiders, let alone use of web spiders by the global update repository. For this additional reason, the 35 U.S.C. § 103 Rejection of Claim 11 is unsupported by the cited art of record and must be withdrawn.

[11] *In re Herrmann, supra.*
[12] Office Action at p. 12.

Additionally, the Applicants made the above argument in the Response mailed November 20, 2007. Considering that the Examiner has not provided any comments or rebuttal to the Applicant's argument, but only restated prior rejections, it can be assumed that the Examiner agrees to the Applicant's arguments and that the Claims are allowable.[13]

Claim 13

Claim 13 recites:

> The method of claim 9, wherein the external update information contains a vendor type, the vendor type being automatic download and release, automatic download and manually confirm release, or manually download and confirm.

The Examiner states:

> ... Microsoft discloses, The method of claim 9, wherein said external update information contains a vendor type, said vendor type being automatic download and release, automatic download and manually confirm release, or manually download and confirm (See section Software Update Service 2.0, p. 13).[14]

The Applicants respectfully disagree. Contrary to the Examiner's statement, Microsoft does not disclose wherein the external update information contains a vendor type, the vendor type being automatic download and release, automatic download and manually confirm release, or manually download and confirm, as required by Claim 13. In support of the Examiner's statement, the Examiner refers to the following portion of Microsoft:

> **Software Update Service 2.0**
> Building upon the strengths of SUS 1.0, the next version will increase administrative flexibility while simplifying overall patch and update management. SUS 2.0 should be available by Spring 2004 and will include numerous improvements and enhancements.
> The initial release of SUS 2.0 will include support the patches available on Microsoft Update, (mentioned above). By the end of 2004, Microsoft plans to include support for all applications on Microsoft Update, (with the possible

[13] *In In re Herrmann, supra.*
[14] Office Action at p. 13.

> exception of MSN and Xbox). Further improvements include: an enhanced reporting environment per machine, per group, or per update information detail; and download and install success or failure reports with error detail.
> SUS 2.0 will support system rollbacks to previous configurations if an installed update causes undesirable results. Also, updates may be targeted to specific machine groups through Active Directory Group Policy or static list-based non–Active Directory definitions.[15]

The cited portion of Microsoft says nothing about a vendor type, let alone that the values of such a vendor type are automatic download and release, automatic download and manually confirm release, or manually download and confirm as required by Claim 13. For this additional reason, the 35 U.S.C. § 103 Rejection of Claim 13 is unsupported by the cited art of record and must be withdrawn.

Additionally, the Applicants made the above argument in the Response mailed November 20, 2007. Considering that the Examiner has not provided any comments or rebuttal to the Applicant's argument, but only restated prior rejections, it can be assumed that the Examiner agrees to the Applicant's arguments and that the Claims are allowable.[16]

In view of the foregoing, it is respectfully asserted that the claims are now in condition for allowance.

Claims 35 and 36

Claim 35 is an *In re Beauregard* claim corresponding to method claim 1. Claim 36 is a non-means-plus-function apparatus claim corresponding to method claim 1. Claim 1 being allowable, Claims 35 and 36 must also be allowable for at least the same reasons as Claim 1.

[15] Microsoft, Software Update Service 2.0, p. 13
[16] *In re Herrmann*, *supra*.

Claims 20-34

Claims 20-34 are means-plus-function claims. In support of the 35 U.S.C. § 103 rejection of Claims 20-34 based on Microsoft White Paper in view of Microsoft Dictionary, the Examiner refers to substantially the same portions of Microsoft used in the rejection of method claims 1-15. According to MPEP Guidelines,

> … Per our holding, the 'broadest reasonable interpretation' that an examiner may give means-plus-function language is that statutorily mandated in paragraph six. Accordingly, *the PTO may not disregard the structure disclosed in the specification corresponding to such language when rendering a Patentability determination* …
> … [The] examiner shall interpret a § 112, 6th paragraph "means or step plus function" limitation in a claim as limited to the corresponding structure, materials or acts described in the specification and equivalents thereof in acts accordance with the following guidelines.[17]

The Guidelines state further:

> … if a prior art reference teaches identity of function to that specified in a claim, then under Donaldson an examiner carries the initial burden of proof for showing that the prior art structure or step is the same as or equivalent to the structure, material, or acts described in the specification which has been identified as corresponding to the claimed means or step plus function.[18]

As Claims 20-34 of the present application are means-plus-function claims and Claims 1-15 of the instant application are non-means-plus-function claims, they cannot be said to be drawn to identical subject matter. Furthermore, the Examiner has not shown for each means-plus-function claim, that the prior art structure or step is the same as or equivalent to the structure, material, or acts described in the specification which has been identified as corresponding to the claimed

[17] "Examination Guidelines For Claims Reciting A "Means or Step Plus Function" Limitation In Accordance With 35 U.S.C § 112, 6th Paragraph," U.S. Patent and Trademark Office, http://www.uspto.gov/web/offices/pac/dapp/pdf/exmgu.pdf, p. 1. (emphasis added)
[18] Guidelines at p. 3. (emphasis in original)

means or step plus function. Therefore, the Examiner has not established a *prima facie* case and the 35 U.S.C. § 103 rejection of Claims 20-34 must be withdrawn.

Additionally, the Applicants made the above argument in the Response mailed November 20, 2007. Considering that the Examiner has not provided any comments or rebuttal to the Applicant's argument, but only restated prior rejections, it can be assumed that the Examiner agrees to the Applicant's arguments and that the Claims are allowable.[19]

Accordingly, a *prima facie* case of obviousness has not been established, and the rejection of claims 1, 20, 35, and 36, and the claims dependent therefrom, based Microsoft White Paper in view of Microsoft Dictionary, is improper.

[19] *In re Herrmann*, *supra*.

Claims Appendix

1. A method for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization, the method comprising:

receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication;

comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers;

queueing the update if an update exists for an application controlled by an inoculation client;

receiving a communication from the corresponding inoculation client checking for available distribution jobs; and

automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client.

2. The method of claim 1, further comprising:

configuring an inoculation server distributed across one or more of the devices; and

performing an initial connection between the inoculation server and the global update repository.

3. The method of claim 1, wherein the application and system information includes operating system information and version.

4. The method of claim 1, wherein the application and system information includes installed software applications and versions.

5. The method of claim 1, wherein the application and system information includes network information.

6. The method of claim 1, wherein the application and system information is received in Extensible Markup Language (XML) format.

7. The method of claim 1, wherein the queuing the update includes linking the update package and the corresponding application in a database table.

8. The method of claim 1, wherein the global update repository is a centralized repository that manages operating systems and software to be delivered to inoculation servers.

9. The method of claim 8, therein the global update repository mines, retrieves, and archives external update information.

10. The method of claim 9, wherein the external update information is mined and retrieved from external security websites.

11. The method of claim 10, wherein the global update repository uses web spiders.

12. The method of claim 1, wherein the comparing includes utilizing an HTTP GET or POST command.

13. The method of claim 9, wherein the external update information contains a vendor type, the vendor type being automatic download and release, automatic download and manually confirm release, or manually download and confirm.

14. The method of claim 1, wherein the comparing is performed by an inventory control engine.

15. The method of claim 1, wherein the queuing is performed by a distribution engine.

16 - 19. (Cancelled)

20. An apparatus for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization, the apparatus comprising:

means for receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication;

means for comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers;

means for queueing the update if an update exists for an application controlled by an inoculation client;

means for receiving a communication from the corresponding inoculation client checking for available distribution jobs; and

means for automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client.

21. The apparatus of claim 20, further comprising:

means for configuring an inoculation server distributed across one or more of the devices; and

means for performing an initial connection between the inoculation server and the global update repository.

22. The apparatus of claim 20, wherein the application and system information includes operating system information and version.

23. The apparatus of claim 20, wherein the application and system information includes installed software applications and versions.

24. The apparatus of claim 20, wherein the application and system information includes network information.

25. The apparatus of claim 20, wherein the application and system information is received in Extensible Markup Language (XML) format.

26. The apparatus of claim 20, wherein the queuing the update includes linking the update package and the corresponding application in a database table.

27. The apparatus of claim 20, wherein the global update repository is a centralized repository that manages operating systems and software to be delivered to inoculation servers.

28. The apparatus of claim 20, therein the global update repository mines, retrieves, and archives external update information.

29. The apparatus of claim 28, wherein the external update information is mined and retrieved from external security websites.

30. The apparatus of claim 29, wherein the global update repository uses web spiders.

31. The apparatus of claim 20, wherein the means for comparing includes means for utilizing an HTTP GET or POST command.

32. The apparatus of claim 28, wherein the external update information contains a vendor type, the vendor type being automatic download and release, automatic download and manually confirm release, or manually download and confirm.

33. The apparatus of claim 20, wherein the means for comparing is an inventory control engine.

34. The apparatus of claim 20, wherein the means for queuing is a distribution engine.

35. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for automatically distributing a software update to a network of devices of a peer-to-peer network controlled by an organization, the method comprising:

receiving application and system information from one or more inoculation clients installed on the devices, the receiving performed via peer-to-peer communication;

comparing the application and system information with application and version information in a global update repository to determine if an update exists for a corresponding application controlled by an inoculation client, the global update repository including updates from multiple application manufacturers;

queueing the update if an update exists for an application controlled by an inoculation client;

receiving a communication from the corresponding inoculation client checking for available distribution jobs; and

automatically transmitting the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client.

36. An apparatus comprising:

a global update repository including software updates from multiple application manufacturers; and

an inoculation server configured to:

receive application and system information from one or more inoculation clients installed on a network of devices of a peer-to-peer network controlled by an organization, the receiving performed via peer-to-peer communication;

compare the application and system information with application and version information in the global update repository to determine if an update exists for a corresponding application controlled by an inoculation client;

queue the update if an update exists for an application controlled by an inoculation client;

receive a communication from the corresponding inoculation client checking for available distribution jobs; and

automatically transmit the update to the corresponding inoculation client in response to the receiving a communication if an update exists for an application controlled by the corresponding inoculation client.

Evidence Appendix

None.

Related Proceedings Appendix

None.

Please charge any additional required fee or credit any overpayment not otherwise paid or credited to our deposit account No. 50-3557.

Respectfully submitted,

NIXON PEABODY LLP

Dated: December 18, 2009

/John P. Schaub/
John P. Schaub
Reg. No. 42,125

NIXON PEABODY LLP
P.O. Box 60610
Palo Alto, CA 94306
Tel. (650) 320-7700
Fax. (650) 320-7701